EXPERIMENT - 30

Object Oriented Programming Lab

Aim

Write a program to illustrate how Template function can be overloaded.

Syeda Reeha Quasar

14114802719

4C7

EXPERIMENT – 30

Aim:
Write a program to illustrate how Template function can be overloaded.

Source Code:

```
#include <iostream>

using namespace std;

template <class T>
void f(T x, T y) { cout << "Template" << endl; }

void f(int w, int z) { cout << "Non-template" << endl; }

int main()
{
    f(1, 2);
    f('a', 'b');
    f(1, 'b');
}
```

Output:

```
PS D:\sem 4\cpp\oops> cd "d:\sem 4\cpp\oops\" ; if ($?) { g++ templateOverLoading.cpp -o templateOverLoading }
; if ($?) { .\templateOverLoading }
Non-template
Template
Non-template
PS D:\sem 4\cpp\oops> []
```

Non-template
Template
Non-template

Viva Questions

Q1). What are templates in C++?

Ans.

Templates are the foundation of generic programming, which involves writing code in a way that is independent of any particular type.

A template is a blueprint or formula for creating a generic class or a function. The library containers like iterators and algorithms are examples of generic programming and have been developed using template concept.
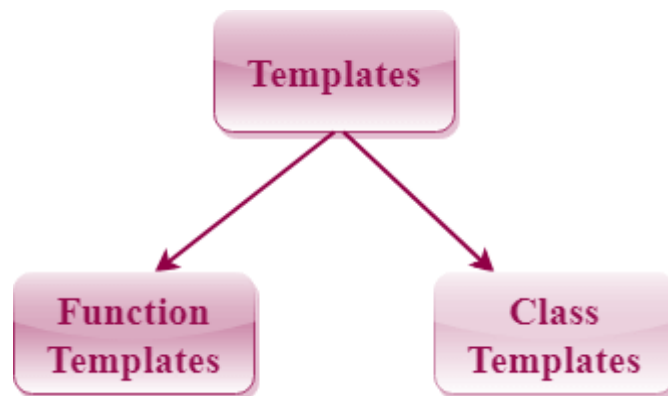
There is a single definition of each container, such as **vector**, but we can define many different kinds of vectors for example, **vector <int>** or **vector <string>**.

Q2). How can templates be classified?

Ans.

Templates can be represented in two ways:

- o Function templates
- o Class templates

Function Templates:

We can define a template for a function. For example, if we have an add() function, we can create versions of the add function for adding the int, float or double type values.

Class Template:

We can define a template for a class. For example, a class template can be created for the array class that can accept the array of various types such as int array, float array or double array.

Q3). Write about Function templates.

Ans.

- o C++ supports a powerful feature known as a template to implement the concept of generic programming.

- o A template allows us to create a family of classes or family of functions to handle different data types.

- o Template classes and functions eliminate the code duplication of different data types and thus makes the development easier and faster.

- o Multiple parameters can be used in both class and function template.

- o Template functions can also be overloaded.

- o We can also use nontype arguments such as built-in or derived data types as template arguments.

Q3) What is purpose of operator overloading?

Ans.

The purpose of operator overloading is to provide a special meaning of an operator for a user-defined data type. With the help of operator overloading, you can redefine the majority of the C++ operators. You can also use operator overloading to perform different operations using one operator.